

Simulátor choreografie roja dronov

Fares Marwan Kamal Bahgat Aboelsaad, Ondrej Babinský,
Michal Přibil, Artemii Kaliadin

Obsah

Obsah	1
1 Katalog Poziadaviek	3
1.1 Uvod	3
1.1.1 Účel katalógu požiadaviek	3
1.1.2 Rozsah využitia simulátora	3
1.1.3 Slovník pojmov	3
1.1.4 Referencie	3
1.1.5 Prehľad nasledujúcich kapitol	4
1.2. Všeobecný popis	5
1.2.1 Perspektíva simulátora	5
1.2.2 Funkcie simulátora	5
1.2.3 Charakteristika používateľov	6
1.2.4 Predpoklady, závislosti a obmedzenia	6
1.3 Špecifikácia požiadaviek	7
1.3.1 Základné definície	7
1.3.2 Nahrávanie súborov	7
1.3.3 Manipulácia s 3D vizualizáciou	7
1.3.3.1 Nastavenie parametrov v konfiguračnom súbore	7
1.3.3.2 Zobrazenie animácie	7
1.3.4 Formát súboru letov	8
1.3.5 Doplnujúce požiadavky	10
2 Navrh	11
2.1 Uvod	11
2.1.1 Účel návrhu	11
2.1.2 Rozsah a zameranie	11
2.2 Formát konfiguračného súboru	12
2.2.1 Gramatika jazyka (ENBF)	12
2.2.2 Štruktúra súboru	12
2.2.3 Definované príkazy	12

2.3. Štruktúra aplikácie	14
2.4. Používateľské rozhranie	16
2.4.1 Hlavná obrazovka simulácie (Simulation Viewer)	16
2.4.1.1 Progress bar	16
2.4.2 Bočný panel (Sidebar)	16
2.4.2.1 Scrollable Content	16
2.4.2.2 Save Settings (Uložiť nastavenia)	17
2.4.3 Obrázky designu	18
2.4.3.1 Hlavný view, zatvorený panel:	18
2.4.3.2 Hlavný view, otvorený panel vyššia časť:	19
2.4.3.3 Hlavný view, otvorený panel nižšia časť:	20
2.5 UML component diagram	21
2.6. UML state diagram	22
2.7. UML sequence diagram	23
2.8. Plán implementácie	24
3 Testovacie scenáre	28
3.1 Úspešné spustenie choreografie s hudbou	28
3.2 Detekcia kolízie pri chybnom letovom pláne	29
3.3 Úprava parametrov simulácie a úspešné spustenie	29
3.4 Použitie tlačidla Restart na resetovanie simulácie	30

1 Katalog Poziadaviek

1.1 Uvod

1.1.1 Účel katalógu požiadaviek

Tento dokument slúži na špecifikáciu požiadaviek pre simulátor choreografie roja dronov pre Steelpark vedy v Košiciach. Tento simulátor bude výstupom projektu v rámci predmetu Tvorba informačných systémov na FMFI UK BA v akademickom roku 2024/2025.

Tento katalóg bol vytvorený na základe požiadaviek od zadávateľa projektu dohodnutých na spoločných stretnutiach. Podrobne popisuje požiadavky na fungovanie pripravovaného simulátoru. Zároveň slúži ako záväzná dohoda medzi zadávateľom a riešiteľmi projektu.

1.1.2 Rozsah využitia simulátora

Cieľom projektu je vyvinúť softvér, sluziaci na simuláciu pohybu dronov pre Steelpark vedy v Košiciach (ďalej "Steelpark"). Tento softvér bude výstupom projektu v rámci predmetu Tvorba informacnych systemov na FMFI UK BA v akademickom roku 2024/2025.

1.1.3 Slovník pojmov

- **Dron** - diaľkovo alebo autonómne (v užšom zmysle len diaľkovo) riadené bezpilotné (najmä motorové) vozidlo (najmä lietadlo a loď, ale často aj pozemné vozidlo)
- **Choreografia dronov**: Koordinovaný pohyb viacerých dronov v trojrozmernom priestore podľa vopred zadaného programu. Každý dron vykonáva sériu pohybov a akcií na základe zadaných inštrukcií, ktoré sú synchronizované s ostatnými dronmi, s cieľom vytvoriť špecifický vizuálny alebo funkčný efekt.

1.1.4 Referencie

- a. GitHub repozitár pripravovaného simulatoru:
<https://github.com/TIS2024-FMFI/drony>
- b. Odkaz na demo: [Unique indoor drones show with a dancer](#)
- c. Použité drony: [CoDrone EDU](#), [DJI Mini 3](#)
- d. Odkaz na SteelPark: <https://steelpark.sk/expozicie/>

1.1.5 Prehľad nasledujúcich kapitol

V kapitole 1.2 všeobecne popisujeme vyvíjaný softvér. Kapitola stručne vysvetľuje funkcionality systému, popisuje charakteristiku a kompetencie

Kapitola 1.3 jasne definuje jednotlivé požiadavky na systém, ktoré boli dohodnuté so zadávateľom projektu.

1.2. Všeobecný popis

1.2.1 Perspektíva simulátora

Vyvíjaný softvér slúži na simuláciu pohybov dronov v 3D prostredí s cieľom umožniť vizualizáciu a testovanie choreografií. Je navrhnutý ako nástroj pre vytváranie a ladenie letových plánov, ktoré môžu byť použité nielen na simuláciu, ale aj pre reálne dronové vystúpenia. Softvér umožňuje používateľom upravovať rozmery priestoru, charakteristiky drona, pričom poskytuje realistický náhľad na výslednú choreografiu.

1.2.2 Funkcie simulátora

- **Načítanie letového plánu a hudby:** Simulátor umožňuje načítať letový plán dronov a hudbu, ktorá bude synchronizovaná so simuláciou. Letový plán definuje presné pohyby dronov, ich pozície a trajektórie.
- **3D vizualizácia:** Simulátor poskytuje 3D vizualizáciu dronov v priestore, ktorá umožňuje používateľovi otáčať, priblížiť a pozrieť si choreografiu z rôznych uhlov. Používateľ má plnú kontrolu nad kamerou a môže sledovať simuláciu z akejkoľvek pozície.
- **Detekcia kolízií:** Ak sa drony počas simulácie zrazia medzi sebou alebo so stenami, simulácia sa zastaví a vizualizuje miesto a čas zrážky. Toto miesto bude zvýraznené červenou farbou, a aplikácia zobrazí aj polohu dronov v čase kolízie.
- **Detekcia vybitia batérie dronov:** V konfiguračnom súbore bude nastavený maximálny čas letu dronov. Ak program letu presiahne tento čas, simulácia sa ani nespustí a používateľovi sa zobrazí upozornenie, že simulácia trvá príliš dlho.
- **Úprava parametrov simulácie:** Používateľ môže meniť rôzne parametre simulácie, ako sú rozmery priestoru (šírka, výška, hĺbka), maximálny čas letu drona, rozmery drona (x, y), a taktiež možnosť zvoliť, ako bude dron vyzeráť: buď z preddefinovaných modelov, alebo nahraním 3D modelu*. Tieto nastavenia umožňujú prispôbiť simuláciu rôznym podmienkam a priestorom.
- **Simulácia so synchronizovanou hudbou:** Simulátor podporuje import hudby, ktorá je synchronizovaná s pohybmi dronov.
- **Štýly letu:** Simulátor podporuje rôzne letové štýly dronov, napríklad presný let alebo let s určitou mierou aproximácie (pomalšie zrýchľovanie alebo plynulejšie zastavenie). Tento parameter môže byť nastavený priamo v letovom pláne.
- **Možnosť manuálnych zásahov počas simulácie (voliteľné, nie je požadované):** Používateľ bude môcť simuláciu kedykoľvek pozastaviť, upraviť letový plán priamo v aplikácii (napr. pomocou textového editora alebo grafického rozhrania), a následne pokračovať so zmenami. Prípadne bude môcť pretočiť animáciu späť.

- **Aplikácia** bude zobrazovať, na ktorom riadku sa simulácia vykonáva v danom čase (voliteľné, nie je požadované).

1.2.3 Charakteristika používateľov

Simulátor bude mať len jeden typ používateľa – bežný používateľ, ktorý bude mať prístup ku všetkým funkciám simulátora.

1.2.4 Predpoklady, závislosti a obmedzenia

- **Predpoklady:** Používatelia budú mať k dispozícii počítač so základnými 3D grafickými schopnosťami (napr. OpenGL, Unity alebo Unreal Engine). Textové súbory obsahujúce letové plány budú pripravené v stanovenom formáte.
- **Závislosti:** Simulátor bude závisieť od externých knižníc alebo softvérového prostredia na zobrazenie 3D grafiky a spracovanie letových plánov.
- **Obmedzenia:** Simulácia bude prebiehať výhradne v interiéri, s obmedzenými výškami a rozmermi priestoru. Presnosť simulácie bude závisieť od zadaných letových plánov a môže byť ovplyvnená obmedzeniami fyzikálneho modelovania.

1.3 Špecifikácia požiadaviek

1.3.1 Základné definície

1. Systém bude realizovaný ako počítačová aplikácia.
2. Systém bude určený pre FullHD display

1.3.2 Nahrávanie súborov

1. Používateľ môže nahráť plán letov vo formáte **.txt**.
2. Používateľ môže nahráť hudbu vo formátoch **.mp3** alebo **.wav**.
3. Používateľ môže nahráť model drona vo formáte **.stl**. (**voliteľné, nie je požadované**)
4. V prípade, že súbor má nesprávny formát, systém upozorní používateľa a požiada ho o nahranie správnej verzie súboru.

1.3.3 Manipulácia s 3D vizualizáciou

1.3.3.1 Nastavenie parametrov v konfiguračnom súbore

1. Používateľ bude musieť nastaviť šírku a výšku priestoru
2. Používateľ bude môcť nastaviť rozmer jednotlivého drona (šírku, výšku, hĺbku)
3. Používateľ bude môcť nastaviť maximálnu dĺžku letu drona
4. Používateľ bude môcť nastaviť 3D model drona, ktorý bude zobrazený počas simulácie. Môže si vybrať z preddefinovaných modelov alebo nahráť svoj vlastný súbor vo formáte **.stl**.*

1.3.3.2 Zobrazenie animácie

1. Rozhranie bude mať tlačidlo **PLAY**, ktoré spustí animáciu pohybu dronov.
2. Rozhranie bude mať tlačidlo **STOP**, ktoré zastaví animáciu pohybu dronov.
3. Rozhranie bude mať tlačidlo **PLAY_REWIND**, ktoré bude spúšťať animáciu späť (voliteľné)
4. Rozhranie zobrazuje číslo riadku, na ktorom v danom čase beží simulácia. (voliteľné)
5. Rozhranie zobrazuje čas behu simulácie.
6. Používateľ môže nastaviť rýchlosť prehrávania simulácie.
7. Každý dron bude zobrazený ako jednoduchý 3D model.

8. Ak používateľ bude chcieť otočiť 3D animáciu, môže to urobiť kliknutím a podržaním ľavého tlačidla myši a následným pohybom myši doľava, doprava, hore alebo dole. Tento pohyb otočí pohľad na simuláciu v rôznych smeroch.
9. Ak bude chcieť používateľ priblížiť alebo vzdialiť pohľad na simuláciu, použije koliesko myši. Rolovaním kolieska dopredu sa pohľad priblíži, zatiaľ čo rolovaním dozadu sa vzdiali.
10. Používateľ bude mať možnosť pohybovať sa po miestnosti (okolo simulácie) pomocou kláves WASD alebo šípok.
11. Hudba bude mať vlastnú časovú os, na ktorej sa zobrazí jej zvuková vlna (waveform) podľa priebehu času. (voliteľné)
12. Pri zobrazení animácie bude špeciálny pivot (playhead) zobrazovať aktuálny čas prehrávania hudby na jej dráhe. (voliteľné)
13. Ak drony narazia, animácia sa zastaví a zobrazí sa miesto a čas zrážky červenou farbou.
14. Ak drony narazia, aplikácia vypíše presnú polohu dronov, ktoré sa zrazili.
15. Ak čas letu presiahne maximálnu dĺžku letu drona, animácia sa zastaví a používateľ bude upozornený.

1.3.4 Formát suboru letov

1. Súbor bude obsahovať definície identifikátorov a postupnosť príkazov pre každý dron. Každý príkaz bude mať unikátny identifikátor (ID_COMMAND), ktorý bude automaticky vygenerovaný systémom.
2. Súbor môže obsahovať komentáre, ktoré sa začínajú symbolom #.
3. Premenné môžu byť definované pomocou kľúčového slova **DEF**, napr.:
 - a. DEF P1 = [1000, 1000, 1000, 70]
 - b. DEF angle1 = 45
4. Premenné s parametrami môžu byť definované pomocou kľúčového slova **DEF(param)**, napr.:
 - a. DEF P2(alpha) = [1000, 1000, 1000, alpha]
5. Súbor bude mať preddefinované príkazy:
 - a. **set-position [x, y, z]** – nastaví pozíciu drona na začiatku simulácie.
 - b. **take-off [height, speed]** – spustí dron a zdvihne ho na výšku „height“ z rýchlosťou „speed“.
 - c. **fly-to [x, y, z, alpha, speed]** – nastaví pozíciu, kam má dron letieť, uhol cieľového otočenia a rýchlosť.
 - d. **fly-spiral [x, y, z, xA, yA, zA, xB, yB, zB, clockWise, number_of_revolutions, speed]** - vytvorí trajektóriu pre let drona v tvare špirály. (x, y, z – cieľový bod; ((xA, yA, zA), (xB, yB, zB)) – body, os, okolo

ktorej bude dron otáčať; clockwise – 1/0, ak ide smerom hodinových ručičiek; number_of_revolutions – počet otočení a speed - rýchlosť).

- e. **fly-circle** [x, y, z, xA, yA, zA, clockwise, number_of_revolutions, speed] - vytvorí trajektóriu pre let drona v tvare kruhu. (x, y, z – cieľový bod; (xA, yA, zA) - stred kruhu, okolo ktorého bude dron otáčať; clockwise – 1/0, ak ide smerom hodinových ručičiek; number_of_revolutions – počet otočení a speed - rýchlosť).
- f.
- g. **fly-sinus, fly-parabola** (voliteľné)
- h. **drone-mode** [exact/approx, exactly/approximate, e/a] – nastaví režim letu drona (presný/ približný).
- i. **fly-trajectory** [[x, y, z, alpha, speed], [x, y, z, alpha, speed], ...] – nastaví trajektóriu, po ktorej dron poletí.
- j. **land** – dron pristane v aktuálnej polohe.
- k. **hover** [time] – dron zostane na aktuálnej pozícii a neprijme ďalší príkaz počas špecifikovaného času.
- l. **hover** [ID_COMMAND] – dron bude čakať, kým nebude dokončený príkaz s daným ID_COMMAND.
- m. **set-color** [r, g, b] – dron zmení farbu na (r, g, b)

6.

7. Každý príkaz bude obsahovať nasledujúce údaje, oddelené čiarkou:

8. [TIMESTAMP, ID_DRON, CMD, CMD_ARGUMENTS]

- a. **TIMESTAMP** - čas v milisekundoch
- b. **ID_DRON** – číselný identifikátor drona.
- c. **CMD** – príkaz (napr. takeoff).
- d. **CMD_ARGUMENTS** – argumenty príkazu (napr. [height]).

9. Štruktúra súboru bude vyzeráť tak:

```
def-drone [length, width, height, 3d model odkaz]
```

```
# definície:
```

```
DEF nieco = [...]
```

```
DEF nieco2 = ...
```

```
# postupnosť príkazov:
```

```
[TIMESTAMP, ID_DRON, CMD, CMD_ARGUMENTS]
```

```
[TIMESTAMP, ID_DRON, CMD, CMD_ARGUMENTS]
```

```
[TIMESTAMP, ID_DRON, CMD, CMD_ARGUMENTS]
```

[TIMESTAMP, ID_DRON, CMD, CMD_ARGUMENTS] [TIMESTAMP, ID_DRON, CMD, CMD_ARGUMENTS] ...

1.3.5 Doplnujúce požiadavky

(Tieto požiadavky sú voliteľné a budú sa realizovať iba v prípade dostatku prostriedkov)

1. Aplikácia umožní editovať animáciu priamo v nej pridaním nových príkazov.
2. Celú animáciu bude možné vytvoriť priamo v aplikácii.
3. V 3D zobrazení bude možné zobraziť trajektórie letu dronov a jednotlivé body.
4. Možnosť nastaviť obrázok na podlahu a tapety v simulátore.
5. Bude doplnkový príkaz, ktorý umožní nastavovať farbu svetielka
6. Vytvoriť aplikáciu pre Android OS, kde bude defaultne ukazovať nejaka predefinovaná simulácia pohybu dronov

2 Navrh

2.1 Uvod

2.1.1 Účel návrhu

Tento dokument slúži ako detailný návrh pre simulátor choreografie roja dronov pre Steelpark vedy v Košiciach. Tento simulátor bude výstupom projektu v rámci predmetu Tvorba informačných systémov na FMFI UK BA v akademickom roku 2024/2025.

Dokument obsahuje všetky informácie potrebné pre pochopenie funkcionality simulátora a jeho implementácie. Obsah tohto dokumentu zahŕňa všetky požiadavky uvedené v *Katalógu požiadaviek*.

2.1.2 Rozsah a zameranie

Pre prácu s týmto dokumentom sa predpokladá znalosť [Katalógu požiadaviek](#), ktorý s týmto návrhom úzko súvisí. Dokument špecifikuje všetky požiadavky uvedené v [Katalógu požiadaviek](#).

2.2 Formát konfiguračného súboru

2.2.1 Gramatika jazyka (ENBF)

```
program = [constants] (command-call | comment) { "\n" (command-call | comment) }
constants = "def" string "=" factor {constants}
command-call = timestamp drone-id command { term }
command = "flyto" | "flyspiral" | "setposition" | "dronemode" | "flytrajectory" |
           "takeoff" | "land" | "hover"
term = factor { ("+" | "-" | "*" | "/") factor }
factor = number
comment = "#" string
```

2.2.2 Štruktúra súboru

Súbor definuje plán choreografie dronov. Každý riadok môže obsahovať

- Komentár, ktorý sa začína symbolom '#',
- Definícia premennej s nasledujúcou štruktúrou:
def <nazov-premennej> = <hodnota-premennej>
- Príkaz s nasledujúcou štruktúrou:
<time-stamp> <drone-id> <command> <command-args>

2.2.3 Definované príkazy

Format definuje nasledujúce príkazy:

- **set-position <x> <y> <z>** – nastaví pozíciu drona na začiatku simulácie.
- **take-off <height>** – spustí dron a zdvihne ho na výšku „height“.
- **fly-to <x> <y> <z> <alpha> <speed>** – nastaví pozíciu, kam má dron letieť, uhol cieľeho otočenia a rýchlosť.
- **fly-spiral <x> <y> <z> <xA> <yA> <zA> <xB> <yB> <zB> <clockWise> <number-of-revolutions> <speed>** - vytvorí trajektóriu pre let drona v tvare špirály. (x, y, z – cieľový bod; x[A/B], y[A/B], z[A/B] – body, os, okolo ktorej bude dron otáčať; clockwise – 1/-1, ak ide smerom hodinových ručičiek; number_of_revolutions – počet otočení, speed - rýchlosť).
- **fly-circle <x> <y> <z> <x0> <y0> <z0> <clockWise> <number-of-revolutions> <speed>** - vytvorí trajektóriu pre let drona v tvare kruha. (x, y, z – cieľový bod; x0, y0, z0 – centralny bod; clockwise – 1/-1, ak ide smerom hodinových ručičiek; number_of_revolutions – počet otočení, speed - rýchlosť).

- **drone-mode <exact/approx>** – nastaví režim letu drona (presný/ približný).
- **fly-trajectory [<x> <y> <z> <alpha> <speed>, ...]** – nastaví trajektóriu, po ktorej dron poletí.
- **set-color <r> <g> ** - nastavi farbu drona podľa rgb
- **land** – dron pristane v aktuálnej polohe.
- **hover <time>** – dron zostane na aktuálnej pozícii a neprijme ďalší príkaz počas špecifikovaného času.
- Voliteľné:
 - **fly-sinus,**
 - **fly-parabola**

2.3. Štruktúra aplikácie

Unity ako framework má svoju vlastnú štruktúru aplikácie. Pre nás bude dôležitým priečinkom priečinkov *Assets*. Tu je stručný prehľad základných priečinkov v adresárovej štruktúre aplikácie pod priečinkom *Assets*:

Scripts/

Obsahuje hlavnú logiku aplikácie vrátane kontrolerov, interpretera, importerov atď.

- **CameraController/**: Obsahuje skript pre posuvanie a otocenie kamery
- **DroneManager/**: Spravuje drony v simulačnom prostredí.
- **Dto/**: Obsahuje DTO's (Data transfer objects)
- **Entities/**: Obsahuje entity
- **Environment/**: Nastavuje a spravuje 3D prostredie a pravidlá simulácie.
- **FileImporter/**: Spracováva import súborov.
- **Interpreter/**: Generuje trajektórie dronov zo súborov letového programu.
- **LogsManager/**: Zapisuje logy do suboru (Windows:
C:\Users\YourUser\AppData\LocalLow\ DefaultCompany\ Drones\GameLogs.txt
MacOS: ~/Library/Application Support/DefaultCompany/Drones/GameLogs.txt)
- **Managers/**: Obsahuje subory, ktoré manazuju entity aplikacie
- **Music/**: Synchronizuje hudbu s choreografiou dronov.
- **SceneManager/**: Spracova scenu
- **UI/**: Spracúva prvky používateľského rozhrania pre interakcie používateľa.
- **Utility/**: Obsahuje utilitne funkcie

Materials/

Skladuje materiály pre drony, prostredie atď.

Scenes/

Ukladá Unity scény.

StandaloneFileBrowser/

Uklada scripty z externej kniznice na nacitanie suborov

StreamingAssets/

Uklada read-only subory, potrebne pocas behu programy

Prefabs/

Ukladá prefaby pre drony, používateľské rozhranie a prvky prostredia.

- **Drones/**
- **UI/**

Resources/

Assety, ktoré budú potrebné počas behu programu.

- **Models/:** 3D modely pre drony a prvky prostredia.
- **Audio/:** Hudobné súbory na synchronizáciu.
- **FlightPrograms/:** Importované letové programy.

2.4. Používateľské rozhranie

2.4.1 Hlavná obrazovka simulácie (Simulation Viewer)

Hlavná stránka aplikácie je **Simulation Viewer**, kde sa cez celý screen zobrazuje 3D priestor simulácie dronov. Kľúčové prvky rozhrania sú:

2.4.1.1 Progress bar

- Nachádza sa v dolnej časti obrazovky a zobrazuje aktuálny stav simulácie v časovej osi.
 - **Aktuálny čas:** Zobrazuje sa na ľavej strane progress baru, uvádza aktuálnu sekundu simulácie.
 - **Koncový čas:** Zobrazuje sa na pravej strane progress baru, uvádza čas, kedy simulácia končí.
- **Ovládacie tlačidlá:** Nachádzajú sa nad progress barom:
 - **Play/Pause:** Slúži na spustenie alebo pozastavenie simulácie.
 - **Stop a restart:** Umožňuje okamžite zastaviť a zároveň presunúť simuláciu na jej začiatok

Na ľavom boku sa nachádza šípka, ktorá kliknutím otvorí sidebar.

2.4.2 Bočný panel (Sidebar)

Na ľavej strane obrazovky sa nachádza **rozbaľovací bočný panel** (Sidebar). Kliknutím na šípku s hover efektom sa otvorí **scrollovateľný sidebar**, ktorý pokrýva väčšinu výšky panelu a umožňuje prístup k nasledujúcim sekciám:

2.4.2.1 Scrollable Content

Obsahuje nasledujúce sekcie nastavení, prístupné cez posuvník (scrollbar):

- **Command File** (Súbor s príkazmi):
 - Možnosť nahráť príkazový súbor pre drony vo formáte **.txt**, jeho nahraním sa importuje config časť daného súboru do nastavení.
- **Simulation Space Size:**

- Nastavenie rozmerov miestnosti zadaním šírky (Width), výšky (Height) a hĺbky (Depth).
- **Playback Speed:**
 - Posuvník (slider) na úpravu rýchlosti prehrávania simulácie. Aktuálna rýchlosť sa zobrazuje nad sliderom.
- **Visual Settings:**
 - Možnosť **Show Audio Visualization**, ktorá upraví progress bar tak, aby zobrazoval vizualizáciu hudby.
- **Wall Settings:**
 - Výber povrchu steny cez **Select Surface** (Floor, Wall, Roof).
 - Nastavenie **Surface Color** (farby povrchu) alebo pridanie **Surface Texture** obrázka.
- **Drone Settings:**
 - Úprava veľkosti dronov pomocou šírky (Width), výšky (Height) a hĺbky (Depth).
 - Nahrať vlastný 3D model drona vo formáte **.sdl**.
 - Výber konkrétnych dronov pomocou checkboxov, pre aplikovanie zmien na jeden alebo viac dronov.

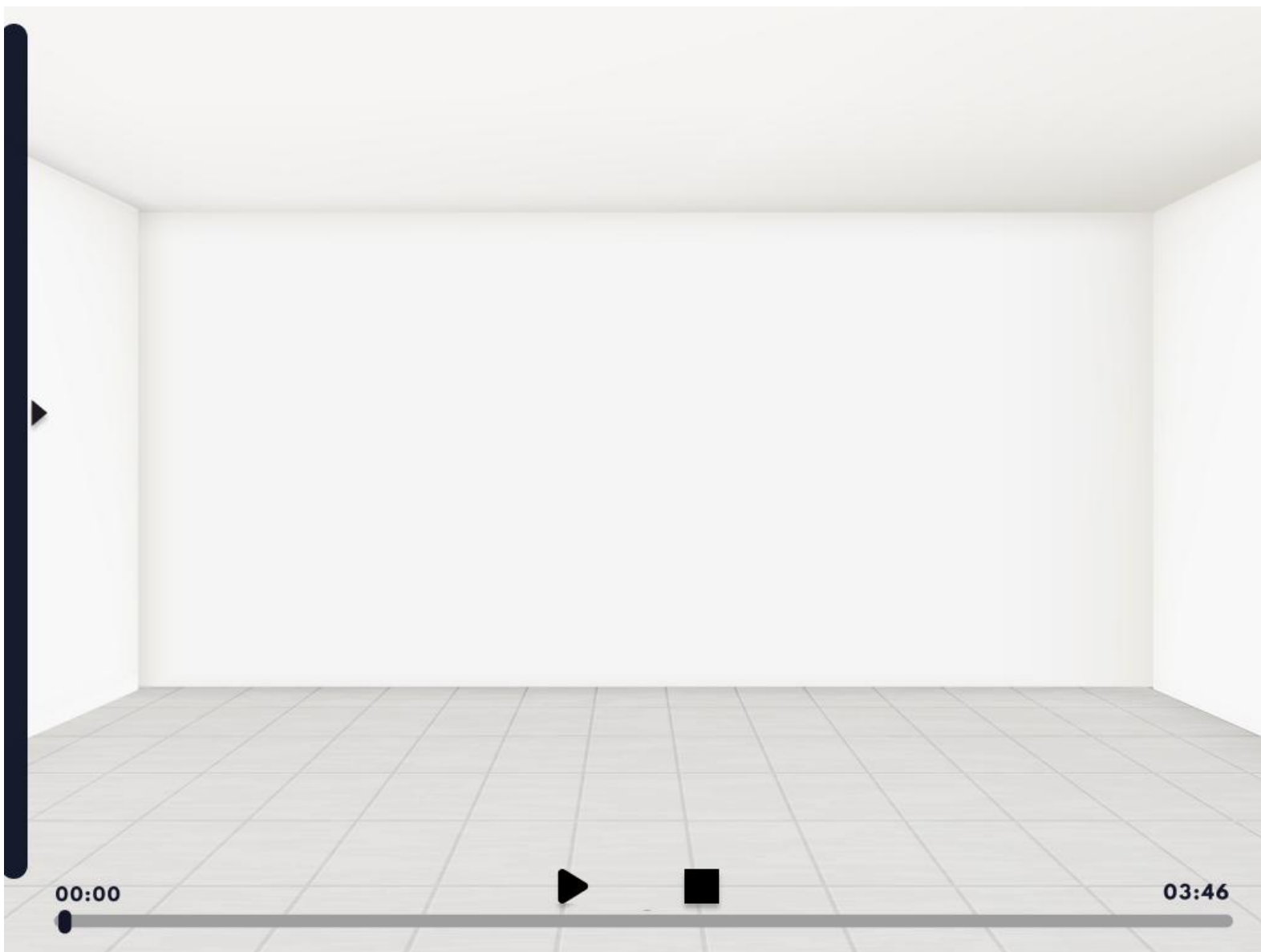
2.4.2.2 Save Settings (Uložiť nastavenia)

- Tlačidlo je pevne umiestnené na spodku sidebaru a umožňuje uloženie všetkých vykonaných zmien nastavení.

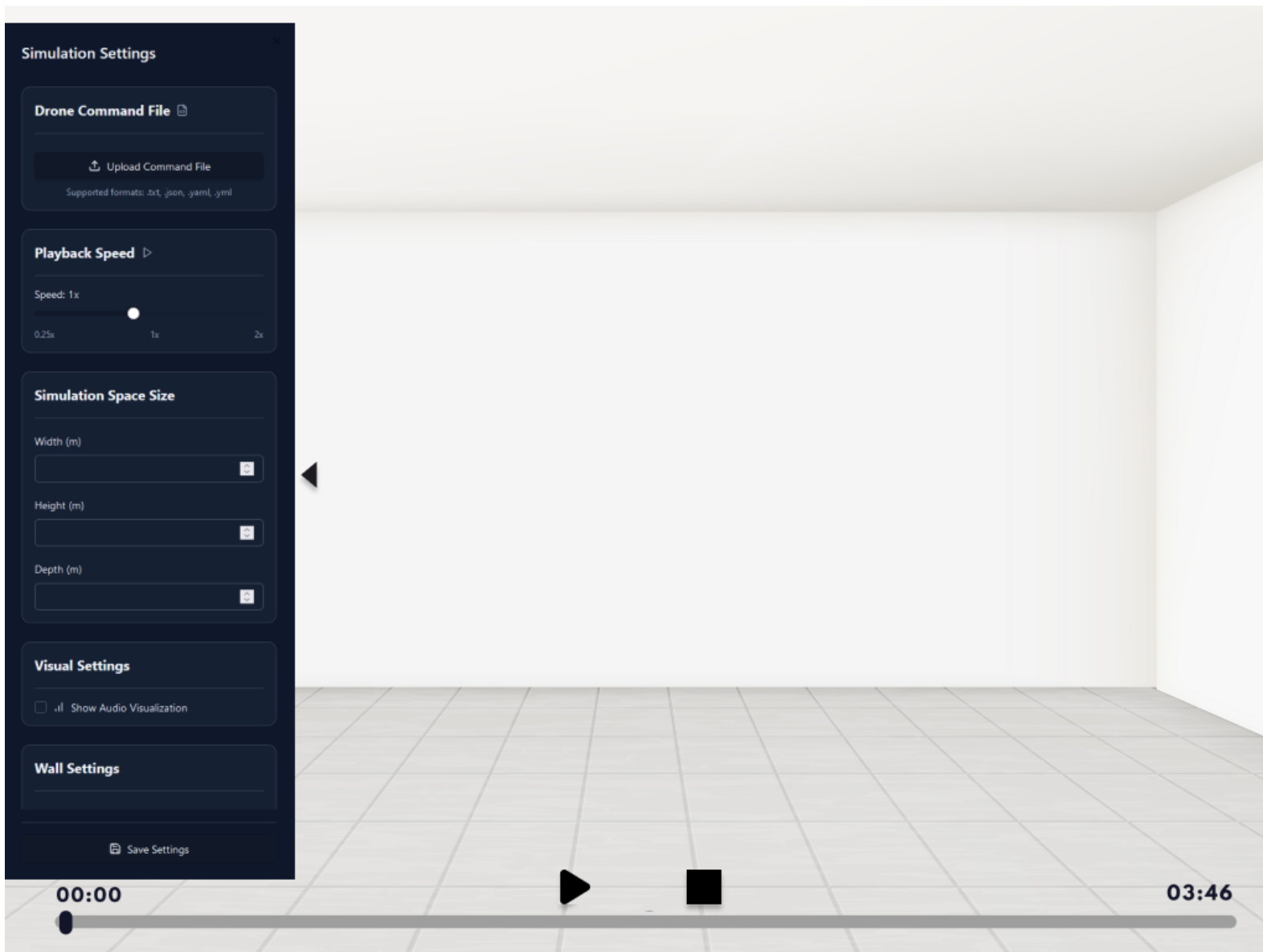
Bočný panel sa dá skryť kliknutím na šípku umiestnenú na jeho pravom okraji.

2.4.3 Obrázky designu

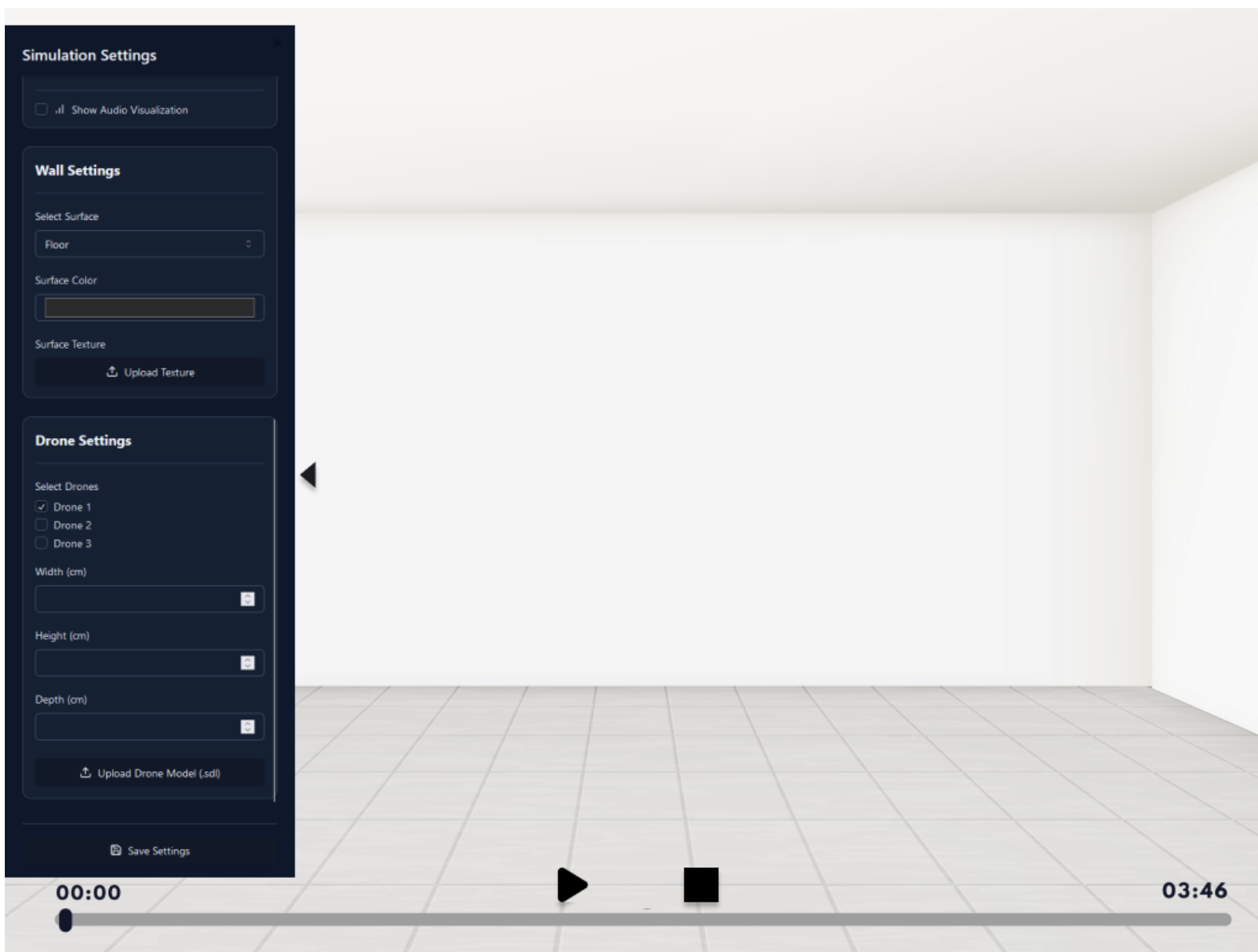
2.4.3.1 Hlavný view, zatvorený panel:



2.4.3.2 Hlavný view, otvorený panel vyššia časť:



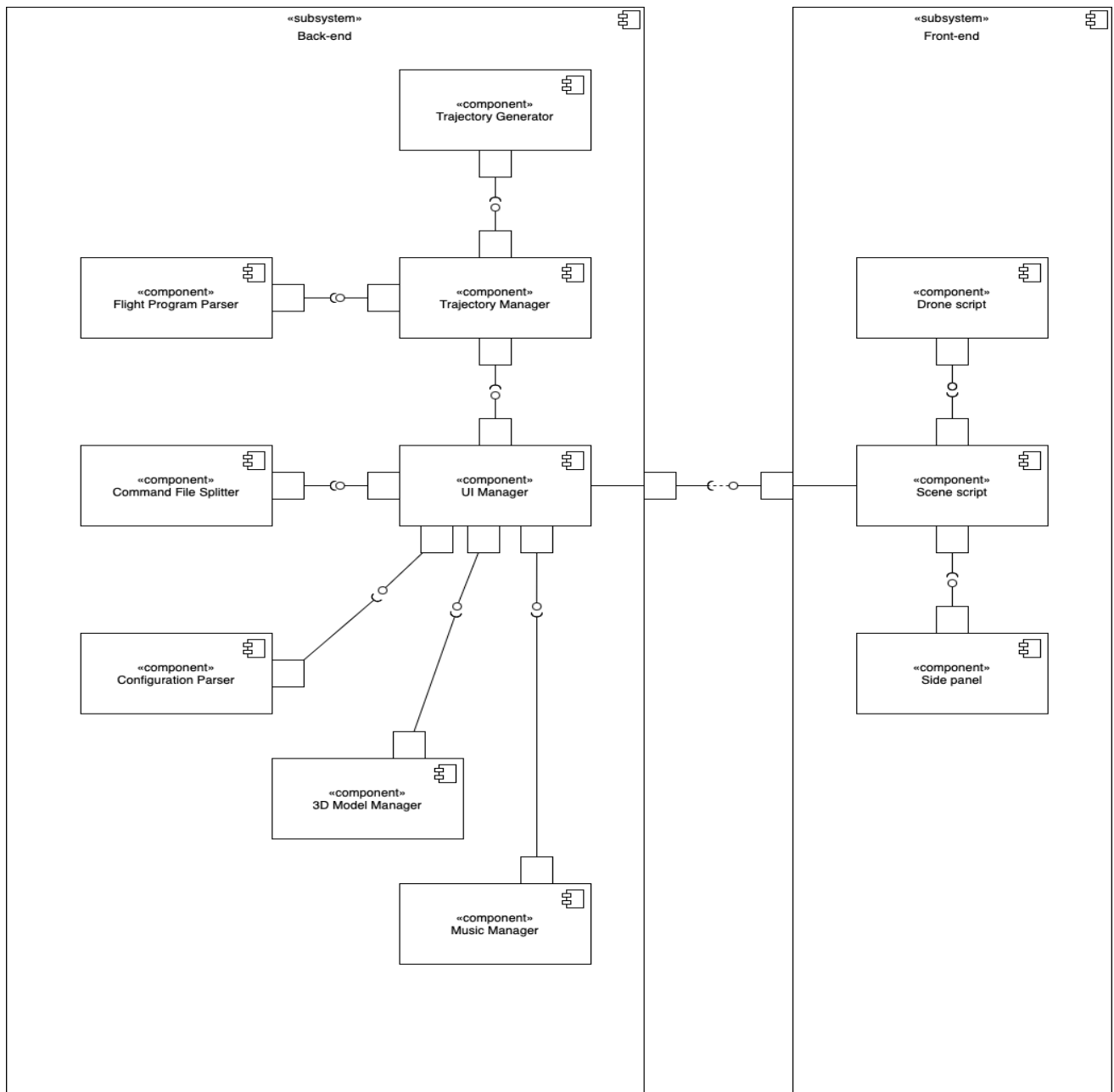
2.4.3.3 Hlavný view, otvorený panel nižšia časť:



2.5 UML component diagram

Tento component diagram, rozdeľuje systém na back-end a front-end.

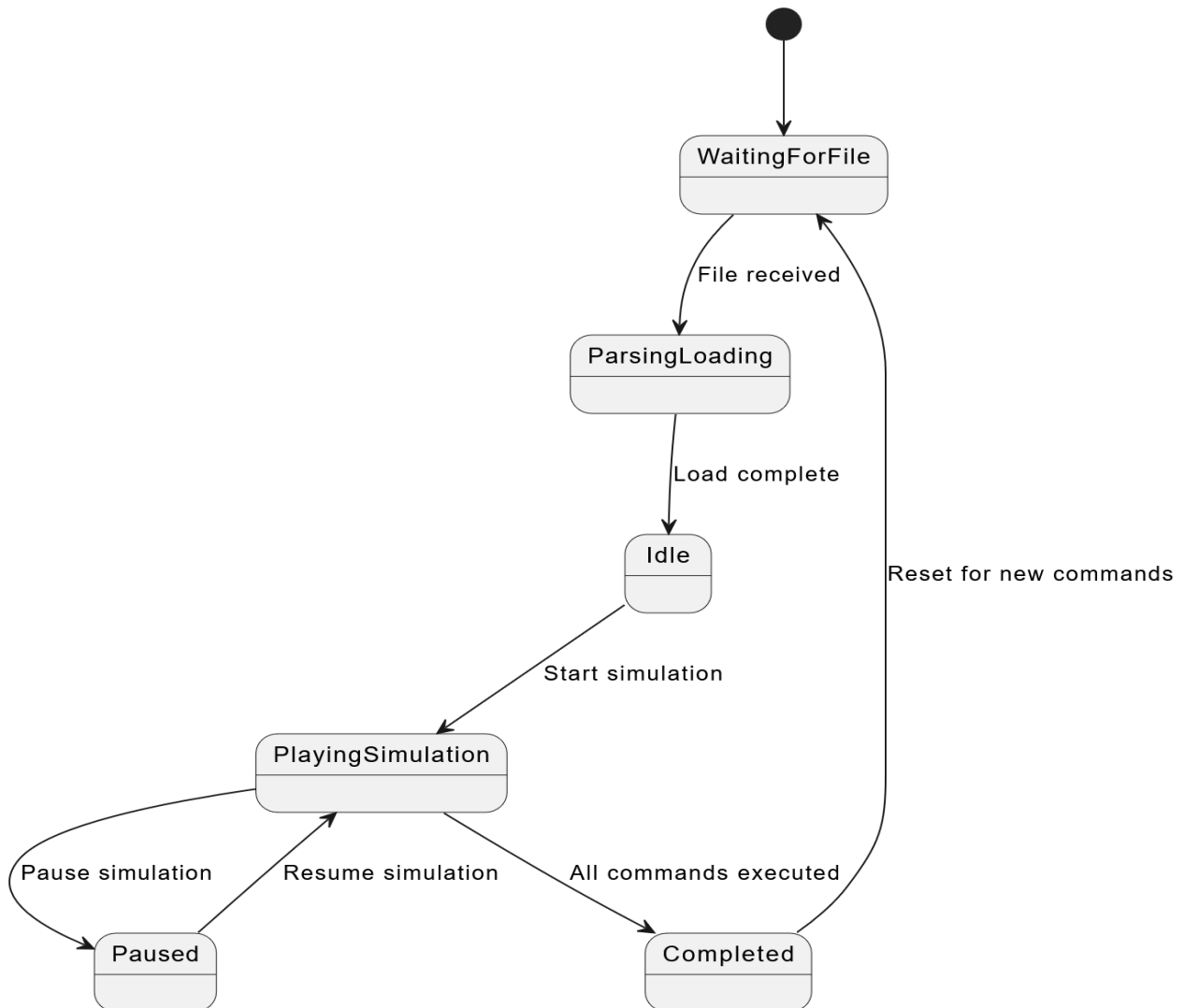
- Back-end spracováva vstupné súbory, rozdeľuje príkazy (Command File Splitter), analyzuje letový program (Flight Program Parser), generuje trajektórie (Trajectory Generator) a spravuje 3D modely a hudbu.
- Front-end riadi vizualizáciu dronov (Drone Script), scénu (Scene Script) a interaktívny bočný panel.
- UI Manager slúži ako hlavný komunikačný bod medzi back-endom a front-endom, zabezpečuje tok dát a vizualizáciu simulácie.



2.6. UML state diagram

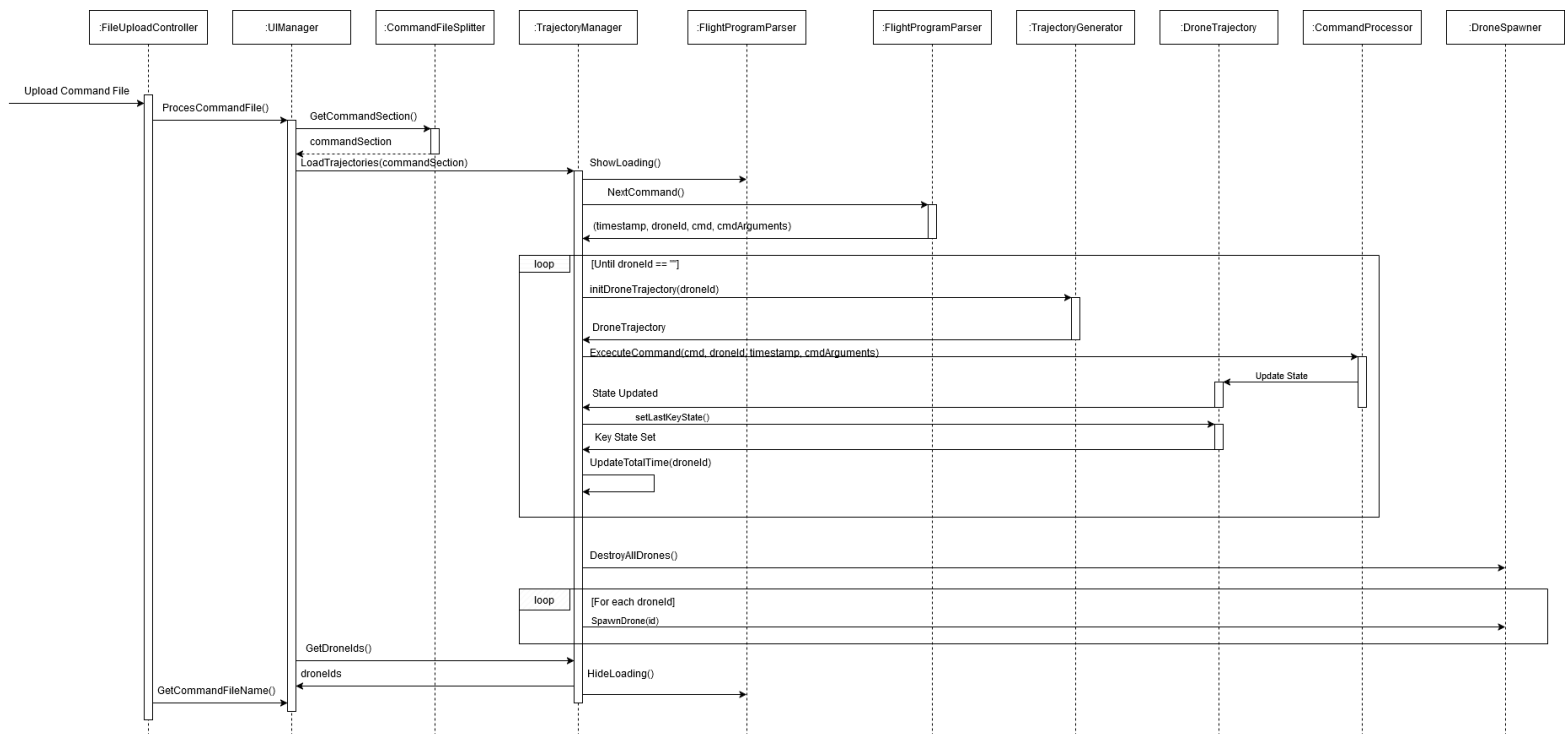
Tento stavový diagram reprezentuje simuláciu letu dronov.

1. Simulácia začína v stave WaitingForFile, kde čaká na vstupný súbor s príkazmi.
2. Po prijatí súboru prechádza do stavu ParsingLoading, kde sa príkazy spracovávajú.
3. Keď je načítanie úspešné, simulácia sa dostane do stavu Idle, kde čaká na spustenie.
4. Po spustení simulácie sa presunie do stavu PlayingSimulation, kde drony vykonávajú svoje naplánované príkazy.
5. Simulácia môže byť pozastavená (Paused) a následne znovu spustená.
6. Po vykonaní všetkých príkazov prechádza do stavu Completed.
7. Nakoniec sa môže simulácia vrátiť späť do WaitingForFile, kde čaká na nový vstup.



2.7. UML sequence diagram

Tento sekvenčný diagram popisuje proces spracovania a vykonania príkazov pre simuláciu letov dronov. Začína sa nahraním súboru s príkazmi, jeho načítaním a rozparsovaním, po ktorom systém iteruje cez jednotlivé drony, generuje ich trajektórie a aktualizuje ich stav, až po dokončenie alebo zastavenie simulácie.



2.8. Plán implementácie

1. Vytvorenie parserov
 - a. Vytvorenie triedy FlightProgramParser pre parsovanie planu letov (Fares Marwan)
 - i. Vytvorenie metody parse (Fares Marwan)
 - b. Vytvorenie triedy ConfigurationParser pre parsovanie konfiguracie simulácie (Fares Marwan)
 - i. Vytvorenie metody parse (Fares Marwan)
 - c. Vytvorenie triedy CommandFileSplitter na rozdelenie suboru prikazov (Fares Marwan)
 - i. Vytvorenie metody split (Fares Marwan)
2. Defenicia entit
 - a. Vytvorenie DroneState entity (Artemii Kaliadin)
 - b. Vytvorenie DroneTrajectory entity (Artemii Kaliadin)
 - c. Defenicia Command enum (Fares Marwan)
3. Vytvorenie triedy Utilities
 - a. Vytvorenie funkcie ConvertFromPlaybackSpeedToMillisGap (Artemii Kaliadin)
 - b. Vytvorenie funkcie OpenFile(filepath) (Ondrej Babinský)
4. Vytvorenie triedy Trajectory Generator
 - a. Vytvorenie metody Generate Empty States (Artemii Kaliadin)
 - b. Vytvorenie metody init Drone Trajectory (Artemii Kaliadin)
 - c. Vytvorenie metody Set Initial Prone Position (Artemii Kaliadin)
 - d. Vytvorenie metody Generate Hover Trajectory (Artemii Kaliadin)
 - e. Vytvorenie metody Generate Linear Trajectory (Artemii Kaliadin)
 - f. Vytvorenie metody Generate Quadratic Bezier Trajectory (Artemii Kaliadin)
 - g. Vytvorenie metody Generate Cubic Bezier Trajectory (Artemii Kaliadin)
 - h. Vytvorenie metody Generate Sinus Trajectory (Michal Přibíl)
 - i. Vytvorenie metody Generate Parabola Trajectory (Michal Přibíl)
5. Vytvorenie triedy Trajectory Manager
 - a. Vytvorenie metody Load Trajectories (Artemii Kaliadin)
 - b. Vytvorenie metody Set Pos Command (Artemii Kaliadin)
 - c. Vytvorenie metody Take Off Command (Artemii Kaliadin)
 - d. Vytvorenie metody Fly To Command (Artemii Kaliadin)
 - e. Vytvorenie metody Drone Mode Command (Artemii Kaliadin)
 - f. Vytvorenie metody Fly Trajectory Command (Artemii Kaliadin)

- g. Vytvorenie metody Fly Spiral Trajectory Command (Artemii Kaliadin)
 - h. Vytvorenie metody Fly Circle Trajectory Command (Artemii Kaliadin)
 - i. Vytvorenie metody Set Color Command (Artemii Kaliadin)
6. Vytvorenie triedy UI Manager
- a. Vytvorenie metody Update Playback Speed (Ondrej Babinský)
 - b. Vytvorenie metody Process Command File (Ondrej Babinský)
 - c. Vytvorenie metody Process Wall Texture (Ondrej Babinský)
 - d. Vytvorenie metody Process Drone Model (Ondrej Babinský)
 - e. Vytvorenie metody Process Music File (Ondrej Babinský)
7. Vytvorenie triedy Music Manager
- a. Vytvorenie metody Process Music File (Michal Přibil)
 - b. Vytvorenie metody Play Music (Michal Přibil)
 - c. Vytvorenie metody Stop Music (Michal Přibil)
 - d. Vytvorenie metody Reset Music (Michal Přibil)
8. Vytvorenie triedy Scene Script (Artemii Kaliadin)
- a. Vytvorenie metody Awake (vytvori instanciu na všetky jednotlivé managery, a posle ich jednotlivým komponentám (napr. droneScript, FileUploadController, atd)
9. Vytvorenie UI
- a. Vytvorenie Side Panelu (Ondrej Babinský)
 - b. Vytvorenie triedy FileUploadController (Ondrej Babinský)
 - i. Vytvorenie metody OnEnable (zavesi callback funkciu z UIManager na každý upload button)
 - ii. Vytvorenie metody OnUploadFile (Ondrej Babinský)
 - c. Vytvorenie warninga pre nesprávny formát súboru (Ondrej Babinský)
 - d. Vytvorenie scriptu pre pohyb kamery (Michal Přibil)
10. Spájanie nahrávanie súborov z interpretom a generátorom trajektórii (cieľ - je spojiť a vytvoriť prvú funkčnú verziu, kde sa dá nahráť jednoduchý súbor, obsahujúci len lineárne pohyby jedného drona, a program otvorí ten súbor, parser prečíta riadky a trajektórie generator vytvorí jednotlivé trajektórie, ktoré potom budú priradené dronu) (Artemii Kaliadin)
- a. Trieda UIManager má funkciu ProcessCommandFile, ktorá je naviazaná ako callback funkcia na proces uploadovania súboru príkazov

- b. Keď používateľ uploadne súbor príkazov (CommandFile), ProcessCommandFile sa zavola a dostane na vstup filepath nahrateho suboru.
- c. Funkcia ProcessCommandFile v triede UIManager otvorí súbor pomocou OpenFile z triedy Utilities, a zavola metódu split v triede CommandFileSplitter
- d. Metóda split rozdelí súbor na dve časti a vráti ich, takže môže raisnúť exception, ak formát nie je správny (ale zatiaľ môžeme to neriesiť, ani pre iné parsery)
- e. Prvá časť zatiaľ bude prázdna, lebo teraz skúsime len generáciu trajektorií
- f. Druhá časť ako string pošle argumentom do metódy LoadTrajectories v triede TrajectoryManager
- g. LoadTrajectories inicializuje FlightProgramParser a bude v cykle volať funkciu NextLine z FlightProgramParsera, ktorý bude už vracať príkaz v formáte [TIMESTAMP, ID_DRON, CMD, CMD_ARGUMENTS]
- h. Potom, keď NextLine vráti null, cyklus skončí a LoadTrajectories vráti True
- i. UIManager dostane ten True a musí reloadnúť scénu z novými trajektoriami

11. Vizualizácia scény

- a. Vytvoriť 3D model pre dron (Michal Přibil)
- b. Pridať do scény gizmos, ktoré budú ukazovať trajektoriu jednotlivých dronov (Artemii Kaliadin)
- c. Zobrazenie farby dronov (Michal Přibil)
- d. Opraviť design miestnosti, Pridať script pre nastavenie rozmerov stien (Michal Přibil)

12. Pridať výnimky (exceptions) pre všetky možné funkcie (ak nie sú ešte tam) a správne ich spracovanie (Fares Marwan)

13. Vytvorenie UI 2

- a. Vytvoriť funkcie pre Play/Stop/Pause (Ondrej Babinský)
- b. Vytvoriť timer (Ondrej Babinský)
- c. Pridať loading UI component. (Lebo asi načítanie súboru bude trvať nejaký čas) (Ondrej Babinský)

14. Detekcia kolízií

- a. Implementovať funkciu detekcie kolízií pre drony (Fares Marwan)
- b. Vytvoriť warning, ktorý bude ukazovať čo sa stalo a v akom čase (Fares Marwan)

15. Nahratie hudby

- a. Vytvorenie funkcie pre nahrávanie hudby (Michal Přibil)
- b. Vytvorenie skriptu pre hudbu, jej prepínanie pri Play/Stop/Pause (Michal Přibil)

16. Vytvorit funkciu pre playback speed (Ondrej Babinský)
17. Implementacia volitelnych poziadaviek (Michal Přibil, Ondrej Babinský, Fares Marwan, Artemii Kaliadin)
18. Testovanie scenarov pouzivania (Ondrej Babinský)
19. Vytvorit auto-mode z predefinovanim planom letov a zbuildnut aplikaciju pre Android OS (Artemii Kaliadin)

3 Testovacie scenáre

3.1 Úspešné spustenie choreografie s hudbou

Požiadavky pokryté: Nahrávanie letového plánu, spustenie simulácie, prehrávanie hudby, ovládanie prehrávania (pozastavenie a pokračovanie), rýchlosť prehrávania.

1.3.1.1, 1.3.2.1, 1.3.3.2.1, 1.3.3.2.5, 1.3.3.2.6, 1.3.3.2.7, 1.3.3.2.8, 1.3.3.2.9, 1.3.3.2.10, 1.3.4.*

Kroky testu:

1. **Nahrat' letový plán:**
 - Klikni na tlačidlo **"Upload command file"** v bočnom paneli.
 - Vyber súbor s správnymi inštrukciami a potvrd' výber.
2. **Nahrat' súbor s hudbou:**
 - Klikni na tlačidlo **"Upload music file"** v bočnom paneli.
 - Vyber súbor s hudbou a potvrd' výber.
3. **Spustiť simuláciu:**
 - V hlavnom okne simulátora stlač tlačidlo **"Play"**.
4. **Sledovať simuláciu:**
 - Sleduj, ako drony vykonávajú choreografiu synchronizovanú s hudbou.
 - Sleduj, v spodnej časti zobrazovanie času simulácie
 - Počas prehrávania stlač tlačidlo **"Pause"**.
 - Over, že simulácia sa pozastavila a drony sa zastavili na mieste.
 - Zmeň v sidepaneli playback speed na 2x
 - Opäť stlač tlačidlo **"Play"**.
 - Over, že simulácia pokračuje od miesta pozastavenia vo väčšej rýchlosti.
 - Vyskúšaj sa pohybovať po priestore myšou, „wasd“ a zoomovať pomocou kolieska.
5. **Dokončiť simuláciu:**
 - Nechaj simuláciu dobehnúť až do konca.
 - Over, že sa úspešne dokončila bez chýb alebo kolízií.

Očakávané výsledky:

- Letový plán sa načíta bez chýb a systém potvrdí úspešné nahratie.
- Simulácia sa spustí a drony vykonávajú choreografiu s hudbou na pozadí.
- Funkcia **"Pause"** správne pozastaví simuláciu, **"Play"** ju obnoví.
- Simulácia prebehne plynulo až do konca bez kolízií.
- Rýchlosť prehrávania sa zmenila na 2x.

3.2 Detekcia kolízie pri chybnom letovom pláne

Požiadavky pokryté: Nahrávanie letového plánu, detekcia kolízií, zobrazenie miesta a času kolízie.

1.3.3.2.13, 1.3.3.2.14

Kroky testu:

1. **Nahrat' letový plán:**
 - Klikni na tlačidlo "**Upload command file**".
 - Vyber súbor s inštrukciami, kde očakávame stret dronov a potvrd' výber.
2. **Spustiť simuláciu:**
 - V hlavnom okne stlač "**Play**".
3. **Overiť detekciu kolízie:**
 - Over, že simulácia sa automaticky zastavila v momente kolízie.
 - Miesto kolízie je zreteľne zvýraznené farbou.
 - Systém zobrazí informácie o kolízií

Očakávané výsledky:

- Letový plán sa načíta bez technických chýb.
- Simulácia sa spustí a drony začnú svoju choreografiu.
- Pri kolízii sa simulácia zastaví a miesto kolízie je vizuálne zvýraznené.
- Systém zobrazí informácie o kolízií

3.3 Úprava parametrov simulácie a úspešné spustenie

Požiadavky pokryté: Nastavenie parametrov simulácie, nahrávanie letového plánu, spustenie simulácie s novými nastaveniami.

3.2.2, 3.2.3, 3.2.4, 3.3.1.1, 3.3.1.2, 3.3.1.4

Kroky testu:

1. **Nahrat' letový plán:**
 - Klikni na tlačidlo "**Upload command file**".
 - Skús vybrať súbor v zlom formáte napríklad .pdf, nebude to dovolené.
 - Vyber súbor s správnymi inštrukciami a potvrd' výber.
2. **Zmeniť veľkosť miestnosti:**
 - Nastav v side paneli novú šírku, výšku a hĺbku miestnosti (napr. zdvojnásob veľkosť).
3. **Zmeniť farbu stien:**

- Vyber novú farbu pre steny.
- 4. **Zmeniť veľkosť drona:**
 - Nastav v side paneli novú šírku, výšku a hĺbku drona (napr. zdvojnásob veľkosť).
- 5. **Spustiť simuláciu:**
 - Stlač tlačidlo **"Play"**.
- 6. **Sledovať simuláciu:**
 - Over, že zmeny sa prejavili (väčšia miestnosť, nová farba stien, väčší dron).
 - Nechaj simuláciu dobehnúť do konca.

Očakávané výsledky:

- Všetky nastavenia sú úspešne aplikované v simulácii.
- Simulácia prebieha hladko s novými parametrami.

3.4 Použitie tlačidla Restart na resetovanie simulácie

Požiadavky pokryté: Ovládanie prehrávania (zastavenie), resetovanie simulácie.

3.3.2.2, 3.3.2.3

Kroky testu:

1. **Nahrat' letový plán:**
 - Klikni na tlačidlo **"Upload command file"** v bočnom paneli.
 - Vyber súbor s správnymi inštrukciami a potvrd' výber.
2. **Použiť tlačidlo Restart:**
 - Po niekoľkých sekundách stlač tlačidlo **" Restart "**.
 - Over, že simulácia sa zastavila a vrátila na začiatok.
3. **Opätovné spustenie simulácie:**
 - Stlač tlačidlo **"Play"**.
 - Over, že simulácia sa spustila od začiatku.
 - Stlač tlačidlo **"Pause"**.
 - Over, že simulácia stojí
 - Stlač tlačidlo **"Play"**.

Očakávané výsledky:

- Tlačidlo **" Restart "** správne zastaví simuláciu a resetuje ju.
- Opätovné stlačenie **"Play"** spustí simuláciu od začiatku bez problémov.
- Tlačidlo **"Pause"** zastaví simuláciu a neskôr ju môžeme znovu spustiť z daného miesta

